

Polynomial Time Algorithm for Graph Isomorphism Testing.

Michael I. Trofimov

September 15, 2011

Abstract

Earlier we introduced (M.I.Trofimov, E.A.Smolenskii, Application of the Electronegativity Indices of Organic Molecules to Tasks of Chemical Informatics, *Russian Chemical Bulletin* 54(2005), 2235-2246.

<http://dx.doi.org/10.1007/s11172-006-0105-6>)

effective recursive algorithm for graph isomorphism testing. In this paper we describe used approach and iterative modification of this algorithm, which modification has polynomial time complexity for all graphs.

Key words: graph isomorphism, NP-complexity, graph invariant, topological index.

1 Introduction.

As 30 years ago [4, 9], in spite of efforts of many researchers, whether the problem of testing of graph isomorphism is NP-complete is still open. There is an interesting explanation of this fact [3]. The authors noted that proofs of NP-completeness seem to require a certain amount of redundancy, which redundancy graph isomorphism problem lacks. For example, in the case of subgraph isomorphism search the same result may be observed even if some edges of given graph will be deleted or some new edges will be added. In contrast, if any edge will be added to one of two isomorphic graphs (or if any edge will be deleted), then the graphs will no longer be isomorphic. So the graph isomorphism problem is not typical NP-complete problem. At present many effective algorithms for graph isomorphism testing were proposed and there are number of investigations with attempts to prove polynomial complexity of some of these algorithms. For example, we note preprint [2].

This article deals with an analysis of complexity of iterative modification of graph isomorphism test algorithm, which had been described earlier [11]. The initial recursive algorithm was developed for effective solution of number of hard tasks of computer chemistry, however, here we will not consider the chemical applications of this approach, we only note that in contrast with many other formal approaches, which are used for graph theory applications in chemistry

today, this approach is based on fundamental physico-chemical notions about structure and properties of a substance.

To describe the algorithm we will use Pascal-like pseudo-code, for example, similar pseudo-code had been used in well-known classical books [1, 13].

2 The general principles of the approach.

Further, without loss of generality of the task, we will consider undirected connected graphs without loops [6, 8]. For the algorithm we will use weights of vertices, which weights may be computed from the following equation [11]:

$$x_i = \frac{1}{d_i + 1} \sum_j x_j + b_i, \quad (1)$$

where d_i is degree of vertex i ;

j is vertex number for all i -adjacent vertices (i.e. vertex j is i -adjacent if an edge (i, j) exists in the graph);

b_i is absolute term, positive number (the cases, where $b_i=0$ will be noted specially).

I.e. the weight of every vertex i is arithmetic mean of i -adjacent vertices weights and initial weight of i th vertex $B_i = (d_i + 1)b_i$. Having written similar equations (1) for all n vertices of the graph, we get the system of n linear equations, with solutions x_1, x_2, \dots, x_n . Every system (1) always has single solution.

In matrix form system (1) can be written as

$$A \cdot \overline{X} = \overline{b},$$

where $A = E - L$;

E is the unity matrix of corresponding size;

$M = \|m_{ij}\|$ is adjacency matrix of the graph;

$L = \|\delta_{ij}\| = N \cdot M$;

$$\delta_{ij} = \begin{cases} \frac{1}{d_i+1} & \text{if } m_{ij} = 1, \\ 0 & \text{otherwise;} \end{cases} -$$

$N = \|\eta_{ij}\|$,

$$\eta_{ij} = \begin{cases} \frac{1}{d_i+1} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Consider the matrix $A^{-1} = \|k_{ij}\|$. Since $\lim_{h \rightarrow \infty} L^h = 0$, we obtain Neumann series:

$$A^{-1} = (E - L)^{-1} = \sum_{h=0}^{\infty} L^h = E + \sum_{h=1}^{\infty} L^h.$$

From that follows:

$$\begin{cases} 1 < k_{ij} < 2 & \text{if } i = j, \\ 0 < k_{ij} < 1 & \text{otherwise.} \end{cases} \quad (2)$$

Two graphs G and G' with the adjacency matrices M and M' , respectively, are isomorphic if and only if there exists the permutation matrix

$$P = \|p_{ij}\|,$$

where $p_{ij} = \begin{cases} 1 & \text{if } i \rightarrow j, \\ 0 & \text{otherwise;} \end{cases}$
such that ([5], p.151):

$$M' = P^{-1} \cdot M \cdot P \quad (3)$$

Note following properties of P :

- 1) P is an orthogonal matrix, i.e. $P^{-1} = P^T$;
- 2) if $G \cong G'$ and $i \rightarrow j$, where i is vertex of G , j is vertex of G' , then $\bar{e}_i = P \cdot \bar{e}_j$ and $E_{ii} = P \cdot E_{jj} \cdot P^{-1}$, where all coordinates of the vector \bar{e}_i are equal to 0 except for the i th coordinate equal to 1 and the element on the crossing of the i th row and i th column of the matrix E_{ii} is equal to 1, while all other elements of E_{ii} are equal to zero.

Let's define \bar{d}_i, \bar{d}'_i vectors with the coordinates d_i, d'_i , respectively. In the matrix form one can write $\bar{d} = M \cdot \bar{e}$ and $\bar{d}' = M' \cdot \bar{e}$, where \bar{e} is unity vector. From that follows $\bar{d}' = P^{-1} \cdot \bar{d}$. Let's define D, D' as the diagonal matrices with the diagonal elements equal to d_i, d'_i , respectively. Then $N = (E + D)^{-1}$ and $N' = (E + D')^{-1}$, where E is the unity matrix of corresponding size. Also D and D' matrices can be expressed through the \bar{d}_i and \bar{d}'_i vectors:

$$D = \sum_i E_{ii} \cdot \bar{d} \cdot \bar{e}_i^T, \quad D' = \sum_j E_{jj} \cdot \bar{d}' \cdot \bar{e}_j^T.$$

Consider two systems of linear equations (1) for G and G' , respectively: $A \cdot \bar{X} = \bar{b}$ and $A' \cdot \bar{X}' = \bar{b}'$.

Assertion 1. *Let $G \cong G'$ and let P be the permutation matrix. Then if $\bar{b}' = P^{-1} \cdot \bar{b}$, then $\bar{X}' = P^{-1} \cdot \bar{X}$.*

Proof. Let's find a relation between the matrices A and A' :

$$P^{-1} \cdot D \cdot P = P^{-1} \cdot \left(\sum_i E_{ii} \cdot \bar{d} \cdot \bar{e}_i^T \right) \cdot P = \sum_i P^{-1} \cdot E_{ii} \cdot \bar{d} \cdot \bar{e}_i^T \cdot P = \sum_i P^{-1} \cdot E_{ii} \cdot P \cdot P^{-1} \cdot \bar{d} \cdot (P^{-1} \cdot \bar{e}_i)^T = \sum_j E_{jj} \cdot \bar{d}' \cdot \bar{e}_j^T = D'.$$

From that follows:

$$N' = (E + D')^{-1} = (P^{-1} \cdot E \cdot P + P^{-1} \cdot D \cdot P)^{-1} = (P^{-1} \cdot (E + D) \cdot P)^{-1} = P^{-1} \cdot (E + D)^{-1} \cdot P = P^{-1} \cdot N \cdot P$$

and also:

$$A' = E - N' \cdot M' = P^{-1} \cdot E \cdot P - P^{-1} \cdot N \cdot P \cdot P^{-1} \cdot M \cdot P = P^{-1} \cdot (E - N \cdot M) \cdot P = P^{-1} \cdot A \cdot P, \quad A'^{-1} = P^{-1} \cdot A^{-1} \cdot P.$$

Finally:

$$\overline{X}' = A'^{-1} \cdot \overline{b}' = P^{-1} \cdot A^{-1} \cdot P \cdot P^{-1} \cdot \overline{b} = P^{-1} \cdot A^{-1} \cdot \overline{b} = P^{-1} \cdot \overline{X},$$

which was to be proved. ■

Assertion 2. For any two graphs G and G' , which graphs have n vertices, and for any permutation $P \in S_n$, if for all \overline{b} , and $\overline{b}' = P^{-1} \cdot \overline{b}$, and for solutions \overline{X} and \overline{X}' of system (1) we have $\overline{X}' = P^{-1} \cdot \overline{X}$, then P is the graphs isomorphism.

Proof. Let M be the adjacency matrix of the graph G , let M' be the adjacency matrix of the graph G' , and M'' be the adjacency matrix of the graph G'' . Let $M'' = P^{-1} \cdot M \cdot P$ and $\overline{b}'' = \overline{b}'$. Then according to definition of graph isomorphism, from $M'' = P^{-1} \cdot M \cdot P$ we have $G \cong G''$. So, according to Assertion 1 from $\overline{b}'' = P^{-1} \cdot \overline{b}$ we have $\overline{X}'' = P^{-1} \cdot \overline{X}$. Taking into account $\overline{X}' = P^{-1} \cdot \overline{X}$, we get $\overline{X}' = \overline{X}''$. Also $A \cdot \overline{X} = \overline{b}$, so from $\overline{b}'' = \overline{b}'$ we have $A'' \cdot \overline{X}'' = A' \cdot \overline{X}'$. Taking into account $\overline{X}' = \overline{X}''$, we have $A' = A''$. This means that $M' = M''$. Taking into account $M'' = P^{-1} \cdot M \cdot P$, finally we have $M' = P^{-1} \cdot M \cdot P$ and $G \cong G'$. ■

It is self-evident that n linear independent vectors \overline{b} are sufficient to correspond to Assertion 2. From that we have following assertion:

Assertion 3. For any two graphs G and G' , which graphs have n vertices, and for any permutation $P \in S_n$, if for n linear independent vectors \overline{b} and for n vectors $\overline{b}' = P^{-1} \cdot \overline{b}$, and for solutions \overline{X} and \overline{X}' of system (1) we have $\overline{X}' = P^{-1} \cdot \overline{X}$, then P is the graphs isomorphism.

According to Harary's definition: "Two points [vertices – MT] u and v of the graph G are *similar* if for some automorphism α of G , $\alpha(u) = v$." [5], p.171. We expand this definition for vertex u of the graph G and vertex v of the graph G' :

Definition 1. If $G \cong G'$ and $v \leftrightarrow u$ is possible mapping, then u and v are *similar*. ■

3 The algorithm.

Let graphs G, G' are isomorphic. For selected pair of vertices (i, j) , $i \in V, j \in V'$ with the same degree we build bigraph $H_{ij} = (V, V', U_{ij})$, where V is set of vertices of graph G ; V' is set of vertices of graph G' ; U_{ij} is set of edges of bigraph H_{ij} , that vertex $p \in V$ connected with vertex $q \in V'$, if $\deg(p) = \deg(q)$ and $k_{ip} = k'_{jq}$. The number of edges in H_{ij} is not more than n^2 . For vectors $\overline{e}_i, \overline{e}_j$ solutions of systems (1) : $\overline{X}_i[p] = \overline{X}'_j[q]$, if there is edge $(p, q) \in U_{ij}$; and $\overline{X}_i[p] \neq \overline{X}'_j[q]$, if there is not edge $(p, q) \in U_{ij}$. Due to (2), every vertex from pair (i, j) in bigraph H_{ij} is incident to only one edge (the edge is edge (i, j)). From (2) we also can see that all $k_{ij} \neq 0$.

Let function $T(U)$ returns a transversal (subset of edges of bigraph $H = (V, V', U)$), if such transversal exists (if there are a few transversals, then the function returns any transversal), and empty set otherwise.

Now we can consider the following procedure:

procedure P1(i, j, R);

1) For input pair of vertices (i, j) , $i \in V$, $j \in V'$ we build bigraph H_{ij} , as described above.

2) For every edge (p, q) from U_{ij} (except edge (i, j)) we build bigraph H_{pq} , as described above. If $T(U_{ij} \cap U_{pq}) \neq \emptyset$, then replace $U_{ij} := U_{ij} \cap U_{pq}$, else remove edge (p, q) from U_{ij} .

3) If $T(U_{ij}) \neq \emptyset$, then found transversal gives isomorphic map for given graphs (the result map will be returned via R). ■

Assertion 4. *If input vertices are similar ($G \cong G'$) then procedure P1 finds isomorphism.*

Proof. 1) Procedure P1 tries to find isomorphism via one-to-one correspondence with vertex weights of given graphs. These weights are defined by functions of initial weights of vertices and so the functions have to be bijective (to exclude degenerate cases), i.e. the necessary condition is one-to-one correspondence with weight of every vertex p and initial weight of every vertex q . Let's prove that used weights correspond to this condition. The weight of vertex p is

$$x_p = \sum_{t=1}^n k_{pt} b_t.$$

Suppose that

$$C_q = \sum_{t \neq q} k_{qt} b_t = \text{const},$$

then

$$x_p = f(b_q) = k_{pq} b_q + C_q.$$

This function is bijective when $k_{pq} \neq 0$. As was noted above, from (2) we get that all $k_{pq} \neq 0$. Hence the result.

2) Let's prove that if in the result of P1 $T(U_{ij}) \neq \emptyset$, then this transversal is single. As was noted above, every vertex from pair (p, q) in bigraph H_{pq} is incident to only one edge and this edge is edge (p, q) . So, for vertices p, q there is only one edge (p, q) in intersection $U_{ij} \cap U_{pq}$, and so after edges replacement $U_{ij} := U_{ij} \cap U_{pq}$ in bigraph H_{ij} : vertex p is incident to only one edge (p, q) and vertex q is incident to only one edge (p, q) . After the loop for all edges (p, q) from U_{ij} (step 2 of P1) we get bigraph H_{ij} , where every vertex is incident to only one edge, so transversal $T(U_{ij})$ is single.

3) Let's prove that if in the result of P1 $T(U_{ij}) \neq \emptyset$, then this transversal maps isomorphism of given graphs. As was noted above: $X_i[p] = X_j[q]$, if there is edge (p, q) in bigraph H_{ij} . In this case $T(U_{ij})$ is common transversal for U_{ij} and for U_{pq} , i.e. for every edge $(u, v) \in U_{ij}$ we can find edge $(u, v) \in U_{pq}$. Consider permutation P , where $p \rightarrow q$, if edge (p, q) exists. For n linear independent vectors $\bar{e}_1, \dots, \bar{e}_n$: let $\bar{b}_p = \bar{e}_p$, $p = 1, \dots, n$; $\bar{b}' = P^{-1} \cdot \bar{b}$. Then we get $\bar{X}' = P^{-1} \cdot \bar{X}$. According to Assertion 3: P is isomorphism.

4) Let in loop 2 of P1 the first n edges of bigraph H_{ij} give isomorphism $\phi_1 : i \leftrightarrow j, r_1 \leftrightarrow r'_1, \dots, r_{n-1} \leftrightarrow r'_{n-1}$. In this case, as was proved (section 2 of this proof): every vertex is incident to only one edge.

Let next step of the loop finds edge (p, q) , where vertices p, q are not similar, and $k_{ip} = k'_{jq}$. Then $T(U_{ij} \cap U_{pq}) = \emptyset$. Otherwise $T(U_{ij} \cap U_{pq})$ would be common transversal for $U_{r_1 r'_1}, \dots, U_{r_{n-1} r'_{n-1}}$, and any isomorphism $p \leftrightarrow q$ would be possible, and so vertices p, q would be similar, which contradicts our assumption.

Let next step of the loop finds edge (p, q) , where vertices p, q are similar. Then if $T(U_{ij} \cap U_{pq}) \neq \emptyset$, then another isomorphism ϕ_2 will be found, otherwise the edge (p, q) will be removed and initial isomorphism ϕ_1 will be saved in result of procedure P1.

Taking into account commutative and associative properties of intersection we have in conclusion that any isomorphism will be found for every search order of loop 2. ■

Now we can consider the following algorithm, where function Verify tests found mapping by procedure P1. The mapping may be tested via equation (3): if the mapping is isomorphism, then the function returns true, otherwise the function returns false.

Algorithm 1.

Input: graphs G, G'

```

if (numbers of vertices of  $G$  and  $G'$  are different)
  or (numbers of edges of  $G$  and  $G'$  are different) then
    goto 1;
Calculate  $A^{-1}, A'^{-1}$ ;
for  $j:=1$  to  $n$  do
  begin
    P1(1,  $j, R$ );
    if Verify ( $G, G', R$ ) then
      begin
        write('  $G \cong G'$  ');
        goto 2;
      end
    end;
1:   write ('  $G \not\cong G'$  ');
2:   ;■

```

Assertion 5. *Algorithm 1 finds isomorphism for given graphs.*

Proof. If graphs G, G' are isomorphic, then there is vertex j ($j = 1, \dots, n$) in graph G' , which vertex j is similar with vertex 1 of graph G . In this case according to Assertion 4 procedure P1 finds isomorphism for input pair $(1, j)$ and function Verify returns true, the algorithm prints a message " $G \cong G'$ " and

stops.

If graphs G, G' are not isomorphic, then if numbers of vertices of G and G' are different or if numbers of edges of G and G' are different, then the algorithm prints a message “ $G \not\cong G'$ ” and stops. Otherwise procedure P1 may find any map R , but function Verify returns false and after n steps of the loop “for $j:=1$ to n do” the algorithm prints a message “ $G \not\cong G'$ ” and stops. ■

Proved Assertion 5 suggests that graph isomorphism problem may be reduced to particular case of common representatives system finding.

4 Complexity of the algorithm.

The complexity of calculation of matrix A^{-1} may be estimated [7] as $O(n^3)$. The number of edges in H_{ij} for procedure P1 is not more than n^2 (see above), hence the complexity of bigraph building may be estimated as $O(n^2)$ and P1 uses not more n^2+1 bigraphs. So the complexity of P1 may be estimated as $O(n^4)$ and the main effort of Algorithm 1 is n calls of P1 in the worst case (function Verify may be implemented much more effective). Hence total complexity is $O(n^5)$.

5 Conclusion.

Side by side with general theoretical result of this work we would like to note its applied result. Sorted solutions of system (1) are graph invariants (topological index). As we can see, the index has high discriminating capability, so it may be useful for many practical tasks outside the graph isomorphism problem [11], sometime with combination with other indices [12], for example, with modifications of Hosoya’s index [10]. The properties of this index discussed here may be useful for these tasks as well. The researches of system (1) had been started in 1987 [14].

Possible corrections and supplemental files will be available via

<http://mt2.comtv.ru/>

and via

<http://webcenter.ru/~mtrofimov/>

The password to unzip exe-files is

hH758-kT402-N3D8a-961fQ-WJL24

6 Acknowledgments.

Many thanks for help to improve this paper to Prof. Eva Tardos (Cornell University), Prof. Peter Grogono (Concordia University, Montreal), Arthur L. Rubin and Emil Jeřábek (Institute of Mathematics, AS CR). Many thanks to Seth Flaxman and to Francesco Cristiano for found bugs in MT2GI program. Special thanks to Richard Cole (New York University) for reading and productive discussion of initial manuscript of this paper. Also special thanks to Andrey

Khalyavin (Moscow State University) for *Assertions 2,3* and productive discussion of the algorithm. Many thanks for advices, replies and constructive criticism of previous article [11], where we described initial principles of our approach, to Reiner Czerwinski (Institut für Softwaretechnik und Theoretische Informatik, TU Berlin), Dan Archdeacon (University of Vermont), Prof. Reinhard Diestel (Hamburg University), Boris Smirnov (N.D.Zelinsky Institute of Organic Chemistry of Russian Academy of Sci., Moscow), Prof. Natalia L. Malinina (Moscow Aviation Institute). Many thanks to Mary P. Trofimov for help in performing the computational experiment and preparation of the manuscript.

References

- [1] A. V. Aho, J. E. Hopcroft, and J. D. Ulman. *Data structures and algorithms*. Addison-Wesley, Massachusetts.
- [2] R. Czerwinski. A polynomial time algorithm for graph isomorphism, 2008. <http://arxiv.org/abs/0711.2010v4>.
- [3] M. R. Garey and D. S. Johnson. *Computers and intractability (A guide to the theory of NP-completeness)*, pages 155–156. W.H. Freeman, San Francisco, 1979.
- [4] G. Gati. Further annotated bibliography on the isomorphism disease. *J. Graph Theory*, 3:95–109, 1979.
- [5] F. Harary. *Graph theory*. Addison-Wesley, Massachusetts, 1969.
- [6] J. Köbler, U. Schöning, and J. Torán. *The graph isomorphism problem - its structural complexity. Prog Theoret Comput Sci*. Birkhäuser, Boston, 1993.
- [7] C. Meyer. Matrix analysis and applied linear algebra, page 119. SIAM, 2000. <http://www.matrixanalysis.com/>.
- [8] G. L. Miller. Graph isomorphism, general remarks. *J. Comput. Syst. Sci.*, 18:128–142, 1979.
- [9] R. C. Read and D. G. Corneil. The graph isomorphism disease. *J. Graph Theory*, 1:339–363, 1977.
- [10] M. I. Trofimov. An optimization of the procedure for the calculation of Hosoya’s index. *J. Math. Chem.*, 8:327–332, 1991.
- [11] M. I. Trofimov and E. A. Smolenskii. Application of the electronegativity indices of organic molecules to tasks of chemical informatics. *Russ. Chem. Bull.*, 54:2235–2246, 2005. <http://dx.doi.org/10.1007/s11172-006-0105-6>.
- [12] M. I. Trofimov and E. A. Smolenskii. The common method for constructing the universal chemical super-indices. In *Computer applications in scientific researches IVTN-2006 session, the proceedings. Moscow*, page 29, Moscow, 2006. http://www.ivtn.ru/2006/pdf/ivtn_2006_proceedings.pdf.

- [13] N. Wirth. *Algorithms + data structures = programs*. Prentice-Hall, Englewood Cliffs and New Jersey, 1976.
- [14] N. S. Zefirov, M. A. Kirpichenok, F. F. Ismaylov, and M. I. Trofimov. Scheme for the calculation of the electronegativities of atoms in a molecule in the framework of Sanderson's principle. *Dokl. Acad. Sci. USSR Chem Proc. New York*, 296:440–443, 1987.